An Online Spectral Learning Algorithm for Partially Observable Nonlinear Dynamical Systems

Byron Boots and Geoffrey J. Gordon

AAAI 2011

Select Lab

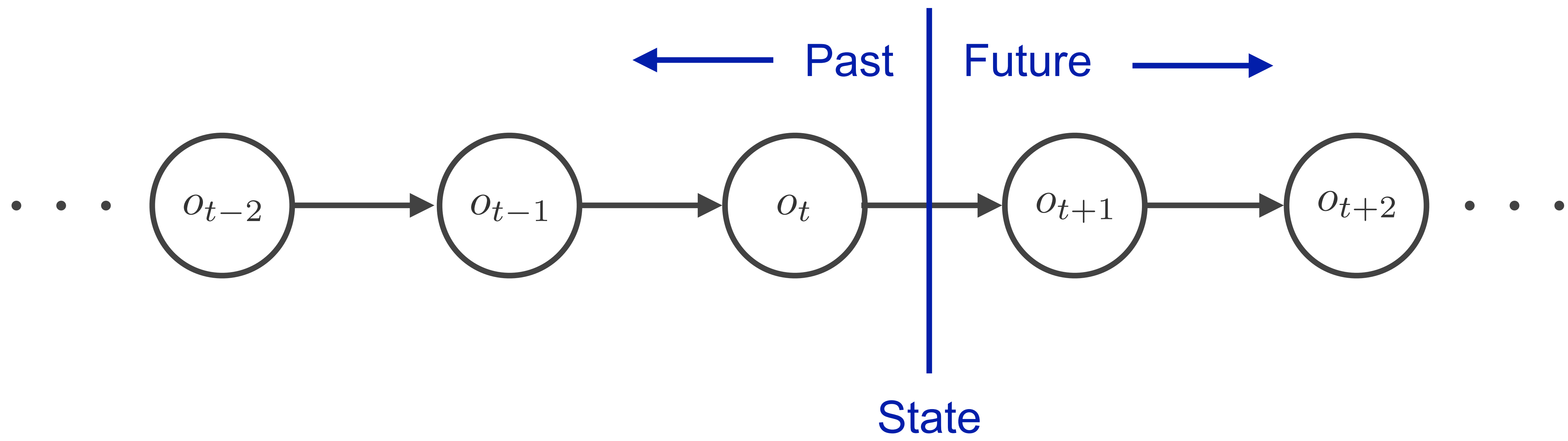
Carnegie Mellon University

What is out there?

$$\cdots \quad o_{t-2} \rightarrow o_{t-1} \rightarrow o_t \rightarrow o_{t+1} \rightarrow o_{t+2} \quad \cdots$$

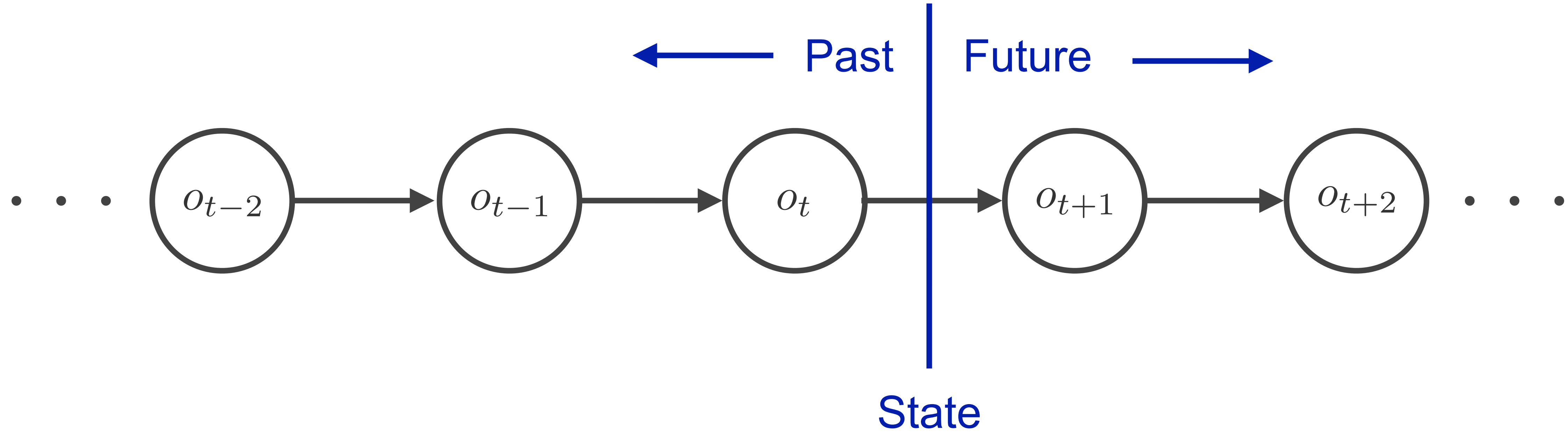
sense
learn
act

What is out there? Dynamical Systems



Dynamical System = A recursive rule for updating state based on observations

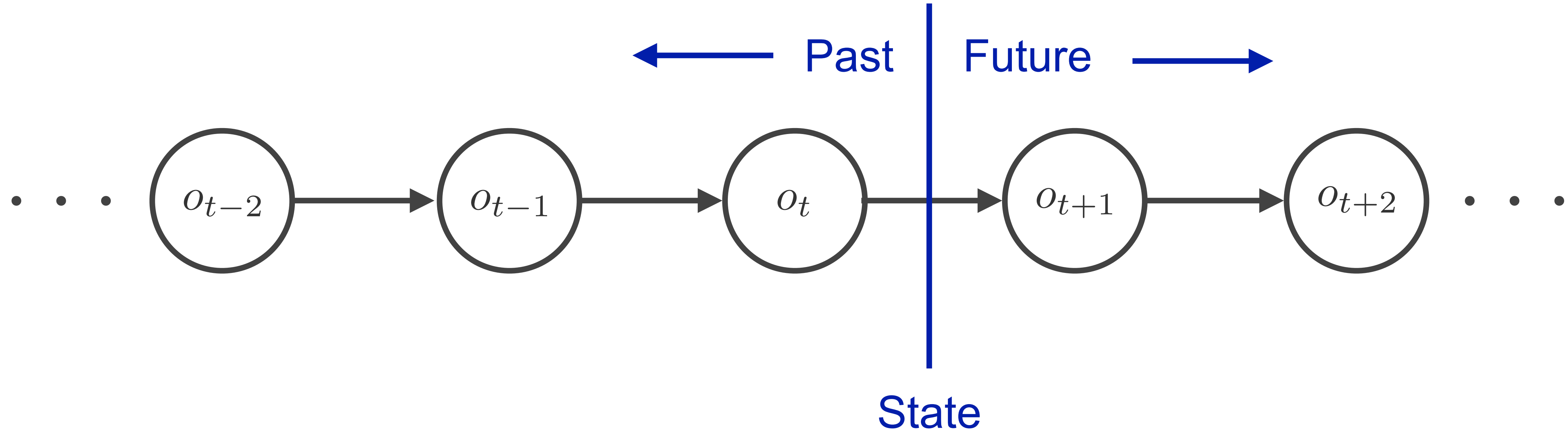
Learning a Dynamical System



we would like to learn a model of a dynamical system

sense
learn
act

Learning a Dynamical System



we would like to learn a model of a dynamical system

today I will focus on Spectral Learning Algorithms for Predictive State Representations sense learn act

Predictive State Representations (PSRs)

comprised of:

set of actions A
set of observations O
initial state $x_1 \in \mathbb{R}^d$
set of transition matrices $M_{ao} \in \mathbb{R}^{d \times d}$
normalization vector $e \in \mathbb{R}^d$

Predictive State Representations (PSRs)

comprised of:

set of actions A
set of observations O
initial state $x_1 \in \mathbb{R}^d$
set of transition matrices $M_{ao} \in \mathbb{R}^{d \times d}$
normalization vector $e \in \mathbb{R}^d$

at each time step can predict:

$$P(o \mid x_t, \mathrm{do}(a_t)) = e^\top M_{a_t,o} x_t$$

sense
learn
act

Predictive State Representations (PSRs)

comprised of:

set of actions A
set of observations O
initial state $x_1 \in \mathbb{R}^d$
set of transition matrices $M_{ao} \in \mathbb{R}^{d \times d}$
normalization vector $e \in \mathbb{R}^d$

at each time step can predict:

$$P(o \mid x_t, \mathrm{do}(a_t)) = e^\top \underline{M_{a_t,o} x_t}$$

Predictive State Representations (PSRs)

comprised of:

set of actions	A
set of observations	O
initial state	$x_1 \in \mathbb{R}^d$
set of transition matrices	$M_{ao} \in \mathbb{R}^{d \times d}$
normalization vector	$e \in \mathbb{R}^d$

at each time step can predict:

$$P(o \mid x_t, \mathrm{do}(a_t)) = \underline{e}^\top M_{a_t,o} x_t$$

Predictive State Representations (PSRs)

comprised of:

set of actions A
set of observations O
initial state $x_1 \in \mathbb{R}^d$
set of transition matrices $M_{ao} \in \mathbb{R}^{d \times d}$
normalization vector $e \in \mathbb{R}^d$

at each time step can predict:

$$P(o \mid x_t, \mathrm{do}(a_t)) = e^\top M_{a_t,o} x_t$$

and update state:

$$x_{t+1} = M_{a_t,o_t} x_t / P(o_t \mid x_t, \mathrm{do}(a_t))$$

Predictive State Representations (PSRs)

comprised of:

set of actions A
set of observations O
initial state $x_1 \in \mathbb{R}^d$
set of transition matrices $M_{ao} \in \mathbb{R}^{d \times d}$
normalization vector $e \in \mathbb{R}^d$

at each time step can predict:

$$P(o \mid x_t, \mathrm{do}(a_t)) = e^\top M_{a_t,o} x_t$$

and update state:

$$x_{t+1} = \underline{M_{a_t,o_t} x_t} / P(o_t \mid x_t, \mathrm{do}(a_t))$$

Predictive State Representations (PSRs)

comprised of:

set of actions A
set of observations O
initial state $x_1 \in \mathbb{R}^d$
set of transition matrices $M_{ao} \in \mathbb{R}^{d \times d}$
normalization vector $e \in \mathbb{R}^d$

at each time step can predict:

$$P(o \mid x_t, \mathrm{do}(a_t)) = e^\top M_{a_t,o} x_t$$

and update state:

$$x_{t+1} = M_{a_t,o_t} x_t / \underline{P(o_t \mid x_t, \mathrm{do}(a_t))}$$

Predictive State Representations

parameters are only determined up to a similarity transform $S \in \mathbb{R}^{d \times d}$

if we replace

$$M_{ao} \to S^{-1} M_{ao} S$$

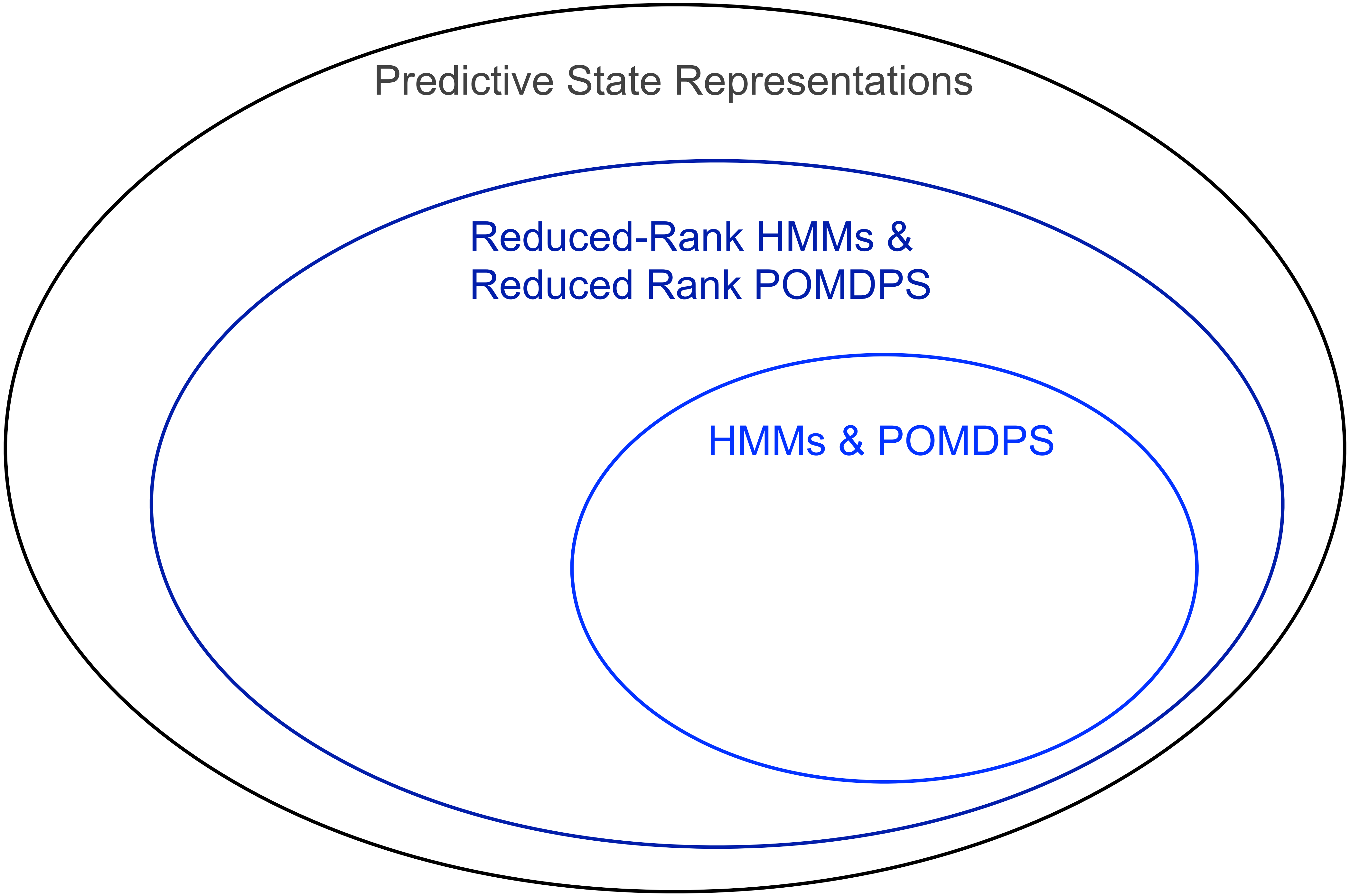
$$x_1 \to S^{-1} x_1$$

$$e \to S^{\top} e$$

the resulting PSR makes exactly the same predictions as the original one

e.g. $P(o \mid x_t, \mathrm{do}(a_t)) = e^{\top} S S^{-1} M_{a_t,o} S S^{-1} x_t$

PSRs Are Very Expressive



for fixed latent dimension *d*

Learning PSRs

can use fast,

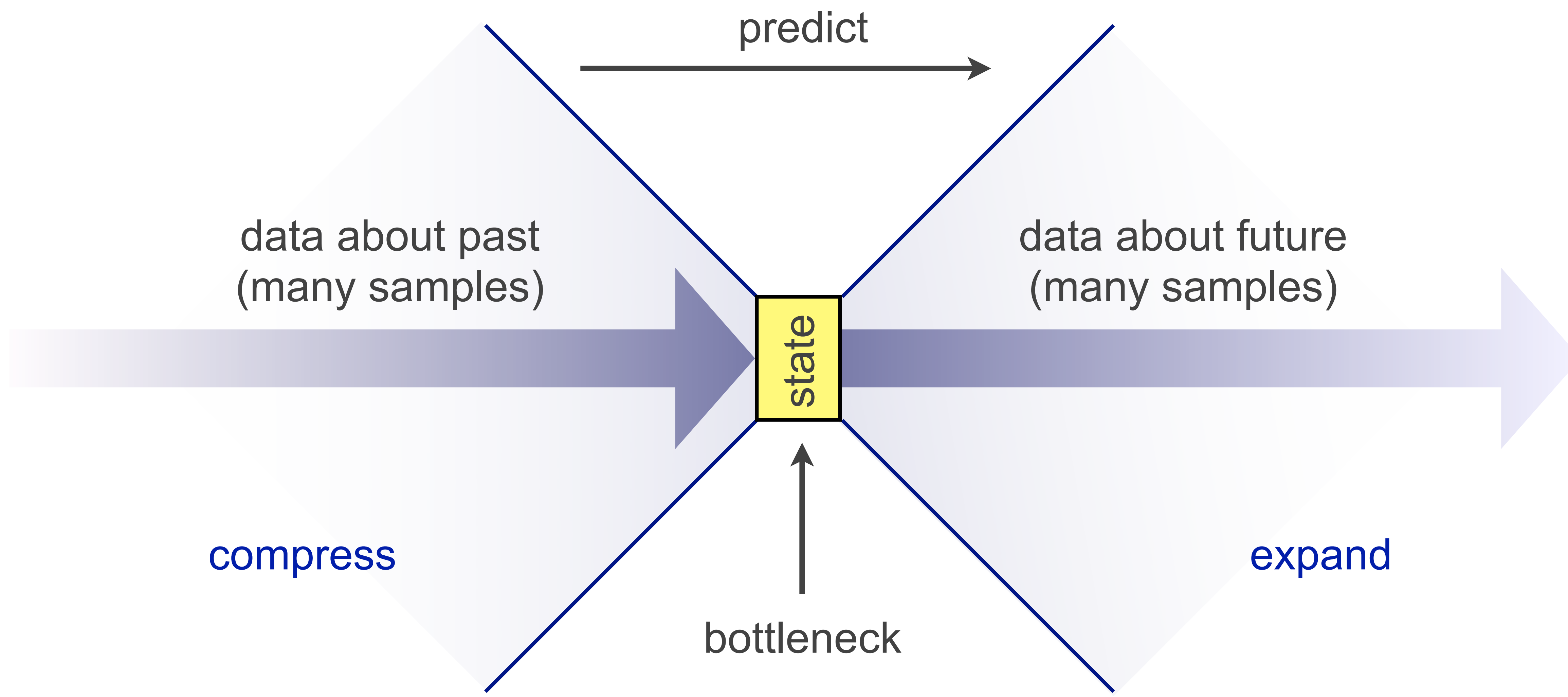
statistically consistent,

spectral methods

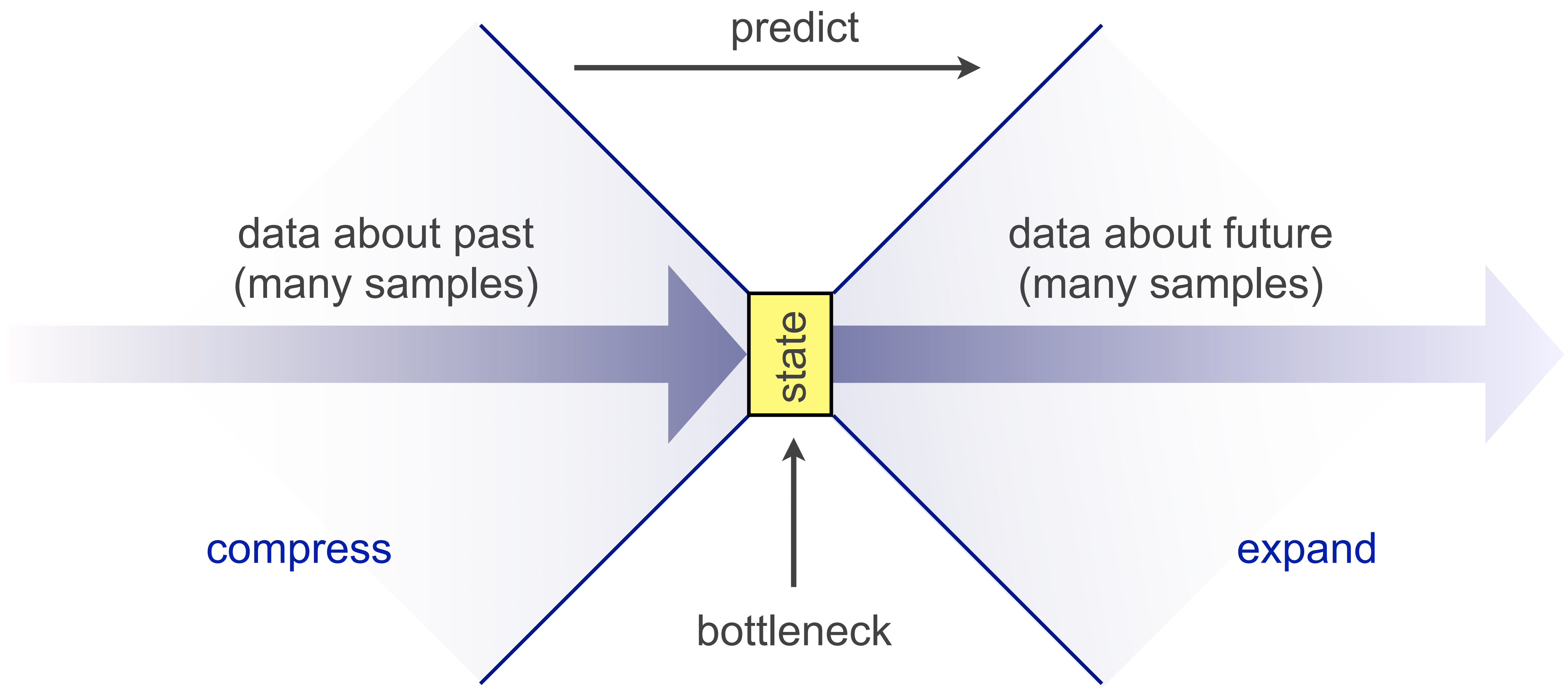
to learn PSR parameters

sense
learn
act

A General Principle



A General Principle



If bottleneck = rank constraint, then get a spectral method

Why Spectral Methods?

There are many ways to learn a dynamical system

- Maximum Likelihood via Expectation Maximization, Gradient Descent, ...
- Bayesian inference via Gibbs, Metropolis Hastings, ...

In contrast to these methods, spectral learning algorithms give

- No local optima:
  - Huge gain in computational efficiency
- Slight loss in statistical efficiency

sense
learn
act

Spectral Learning for PSRs

moments of directly observable features

$\Sigma_{\mathcal{T},\mathcal{AO},\mathcal{H}}$ "trivariance" tensor of features of the future, present, and past

$\Sigma_{\mathcal{T},\mathcal{H}}$ covariance matrix of features of the future and past

$\Sigma_{\mathcal{AO},\mathcal{AO}}$ covariance matrix of features present

Spectral Learning for PSRs

moments of directly observable features

$\Sigma_{\mathcal{T},\mathcal{AO},\mathcal{H}}$ "trivariance" tensor of features of the future, present, and past

$\Sigma_{\mathcal{T},\mathcal{H}}$ covariance matrix of features of the future and past

$\Sigma_{\mathcal{AO},\mathcal{AO}}$ covariance matrix of features present

U left *d* singular vectors of $\Sigma_{\mathcal{T},\mathcal{H}}$

Spectral Learning for PSRs

moments of directly observable features

$\Sigma_{\mathcal{T},\mathcal{AO},\mathcal{H}}$ "trivariance" tensor of features of the future, present, and past

$\Sigma_{\mathcal{T},\mathcal{H}}$ covariance matrix of features of the future and past

$\Sigma_{\mathcal{AO},\mathcal{AO}}$ covariance matrix of features present

U left *d* singular vectors of $\Sigma_{\mathcal{T},\mathcal{H}}$

$$S^{-1}M_{ao}S \; := \; \Sigma_{\mathcal{T},\mathcal{AO},\mathcal{H}} \times_1 U^\top \times_2 \phi(ao)^\top (\Sigma_{\mathcal{AO},\mathcal{AO}})^{-1} \times_3 (\Sigma_{\mathcal{T},\mathcal{H}}^\top U)^\dagger$$

the other parameters can be found analogously

sense
learn
act

Spectral Learning for PSRs

Spectral Learning Algorithm:

- Estimate $\Sigma_{\mathcal{T},\mathcal{AO},\mathcal{H}}$, $\Sigma_{\mathcal{T},\mathcal{H}}$, and $\Sigma_{\mathcal{AO},\mathcal{AO}}$ from data
- Find $\widehat{U}$ by SVD
- Plug in to recover PSR parameters

- Learning is Statistically Consistent
- Only requires Linear Algebra

For details, see:

B. Boots, S. M. Siddiqi, and G. Gordon. *Closing the learning-planning loop with predictive state representations*. RSS, 2010.

Infinite Features

- Can extend the learning algorithm to infinite feature spaces
  - Kernels

- Learning algorithm that we have seen is linear algebra
  - works just fine in an arbitrary RKHS
  - Can rewrite all of the formulas in terms of Gram matrices
  - Uses kernel SVD instead of SVD

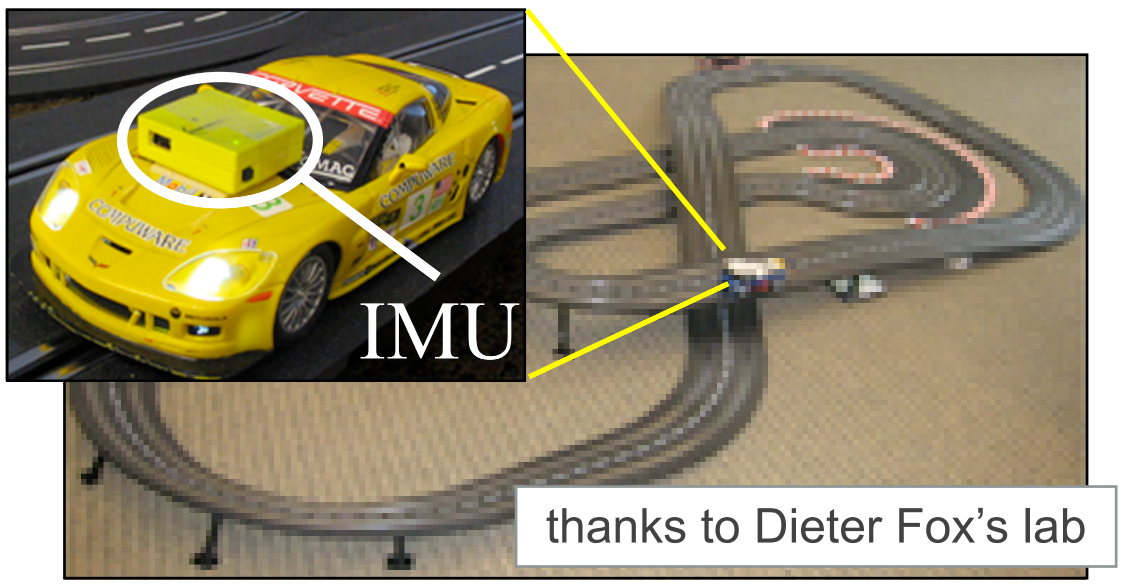
Result: Hilbert Space Embeddings of Dynamical Systems

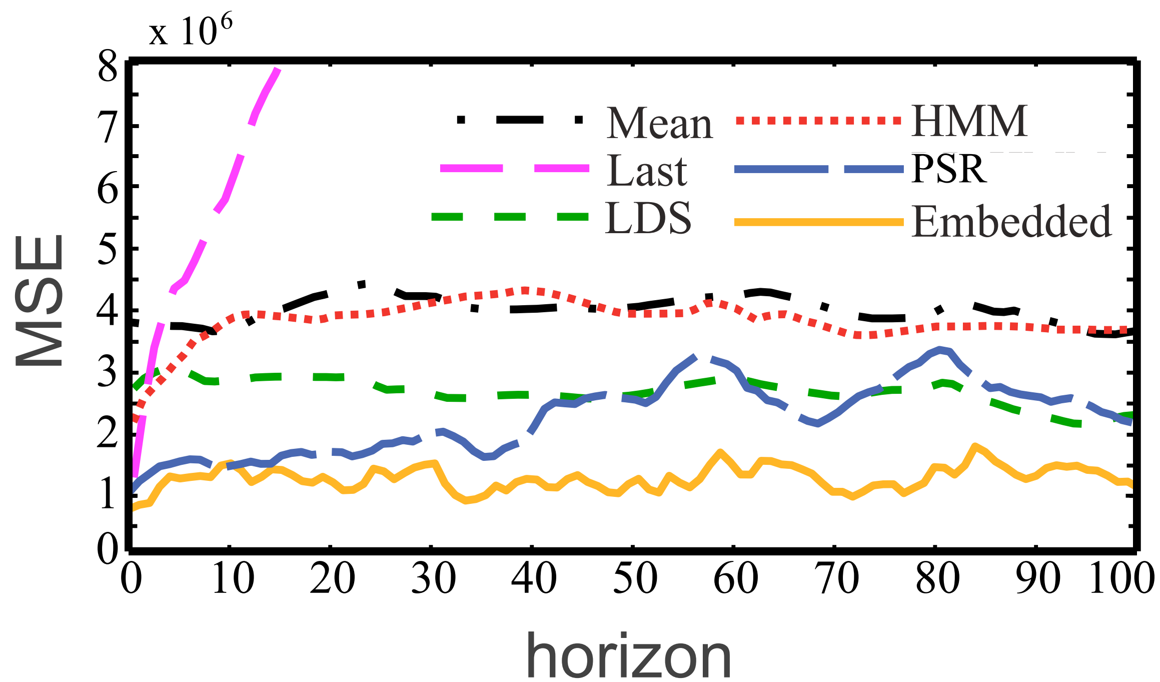
- handles near arbitrary observation distributions
- good prediction performance

For details, see:

L. Song, B. Boots, S. M. Siddiqi, G. Gordon, and A. J. Smola. *Hilbert space embeddings of hidden Markov models*. ICML, 2010.

An Experiment





L. Song, B. Boots, S. M. Siddiqi, G. Gordon, and A. J. Smola. *Hilbert space embeddings of hidden Markov models*. ICML, 2010.

Batch Methods

- Bottleneck: SVD of Gram or Covariance matrix
  - G: (# time steps)^2
  - C: (# features × window length) × (# time steps)

- E.g., 1 hr video, 24 fps, 300×300, features of past and future are all pixels in 2 s windows
  - G: $(3600 \times 24) \times (3600 \times 24) \approx 10^{10}$

sense
learn
act

Making it Fast

- Two techniques
  - online learning
  - random projections
- Neither one new, but combination with spectral learning for PSRs is, and makes huge difference in practice

sense
learn
act

Online Learning

U left *d* singular vectors of $\Sigma_{\mathcal{T},\mathcal{H}}$

$$S^{-1}M_{ao}S \ := \ \Sigma_{\mathcal{T},\mathcal{AO},\mathcal{H}} \times_1 U^\top \times_2 \phi(ao)^\top (\Sigma_{\mathcal{AO},\mathcal{AO}})^{-1} \times_3 (\Sigma_{\mathcal{T},\mathcal{H}}^\top U)^\dagger$$

Online Learning

U left *d* singular vectors of $\Sigma_{\mathcal{T},\mathcal{H}}$

$$S^{-1} M_{ao} S \;:=\; \Sigma_{\mathcal{T},\mathcal{AO},\mathcal{H}} \times_1 U^\top \times_2 \phi(ao)^\top (\Sigma_{\mathcal{AO},\mathcal{AO}})^{-1} \times_3 (\Sigma_{\mathcal{T},\mathcal{H}}^\top U)^\dagger$$

- With each new observation, rank-1 update of:
  - SVD (Brand)
  - inverse (Sherman-Morrison)
- *n* features; latent dimension *d*; *T* steps
  - space = O(*nd*): may fit in cache!
  - time = O(nd^2T): bounded time per example

Random Projections

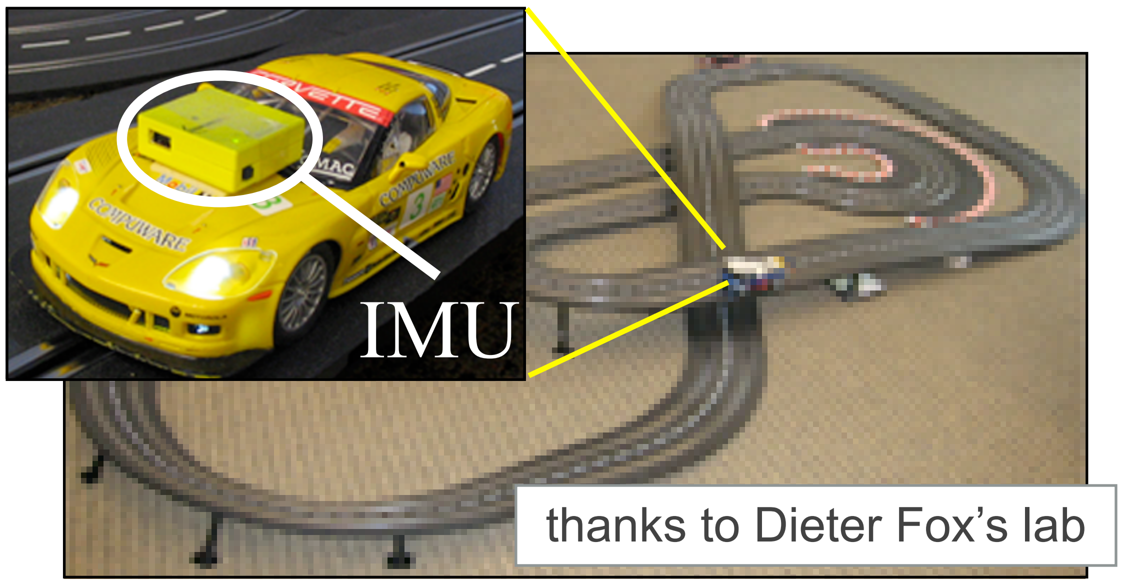
U left *d* singular vectors of $\Sigma_{\mathcal{T},\mathcal{H}}$

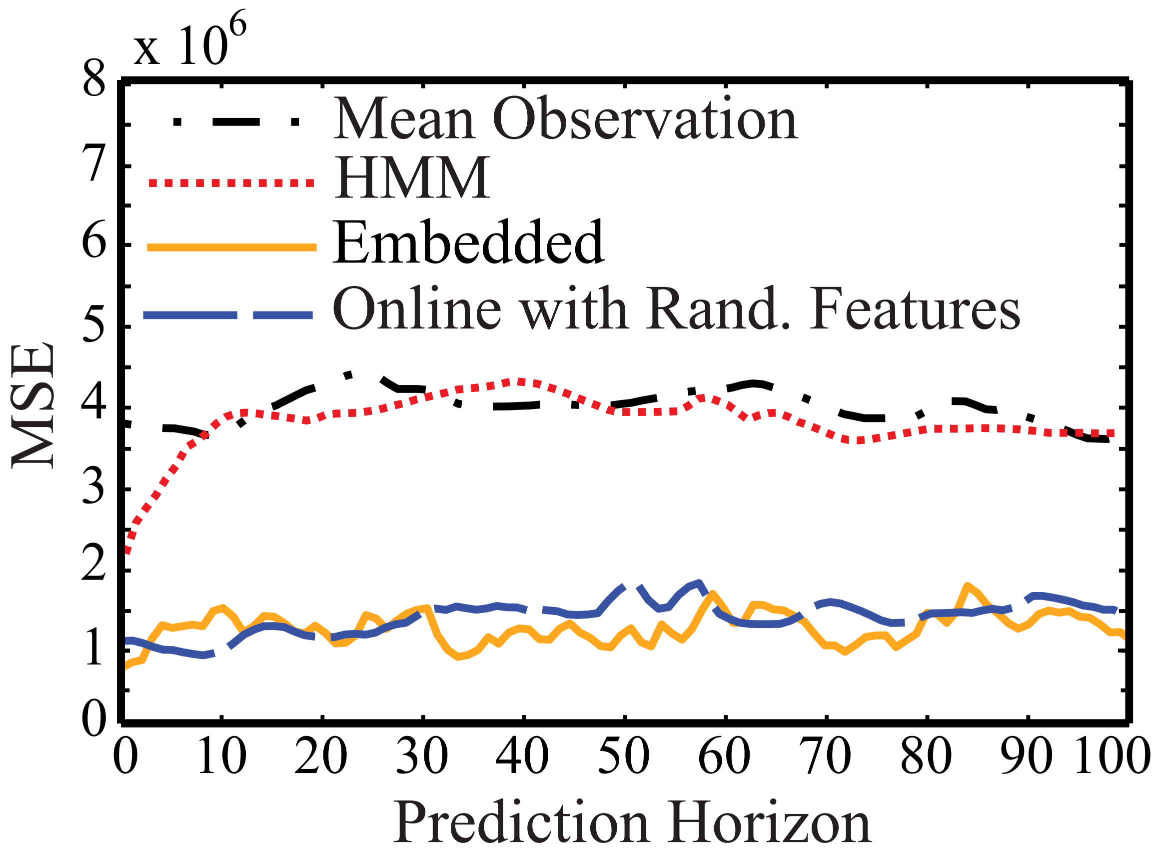
$$S^{-1}M_{ao}S \ := \ \Sigma_{\mathcal{T},\mathcal{AO},\mathcal{H}} \times_1 U^\top \times_2 \phi(ao)^\top \underline{(\Sigma_{\mathcal{AO},\mathcal{AO}})^{-1}} \times_3 (\Sigma_{\mathcal{T},\mathcal{H}}^\top U)^\dagger$$

- With each new observation, rank-1 update of:
  - SVD (Brand)
  - inverse (Sherman-Morrison)
- *n* features; latent dimension *d*; *T* steps
  - space = O(*nd*): may fit in cache!
  - time = O(nd^2T): bounded time per example
- Problem: no rank-1 update of kernel SVD!
  - can use random projections *[Rahimi & Recht, 2007]*

sense
learn
act

Experiment (Revisited)

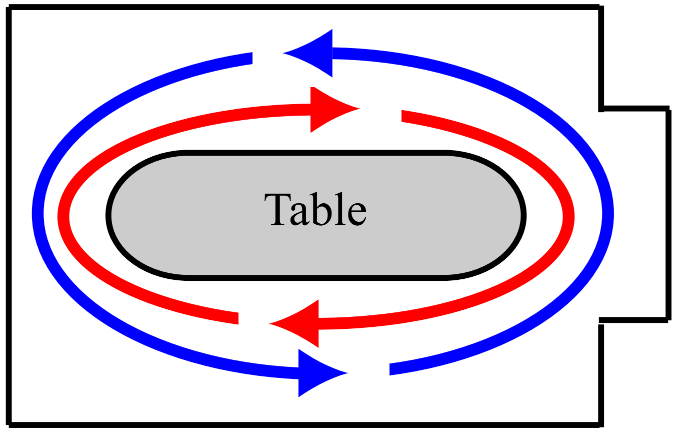




sense
learn
act

Online Learning Example

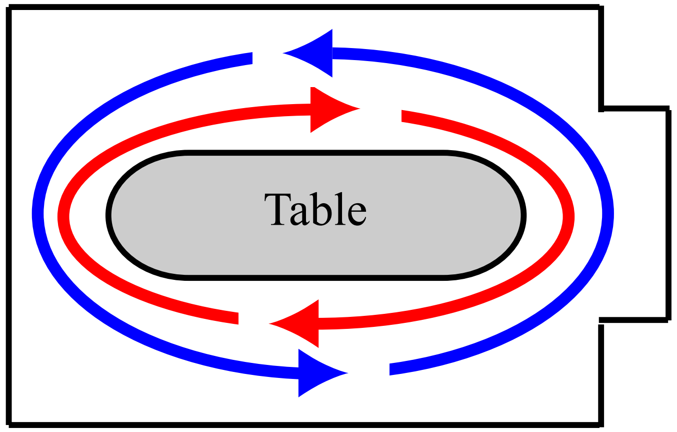
Conference Room



sense
learn
act

Online Learning Example

Conference Room

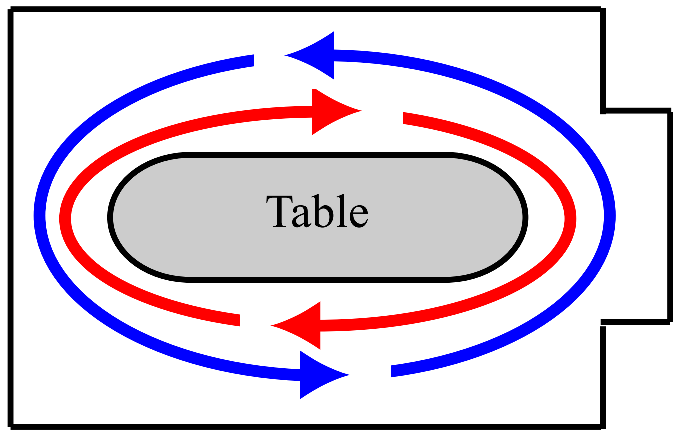


- online+random: 100k features, 11k frames, limit = avail. data
- offline: 2k frames, compressed & subsampled, compute-limited

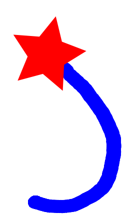
sense
learn
act

Online Learning Example

Conference Room



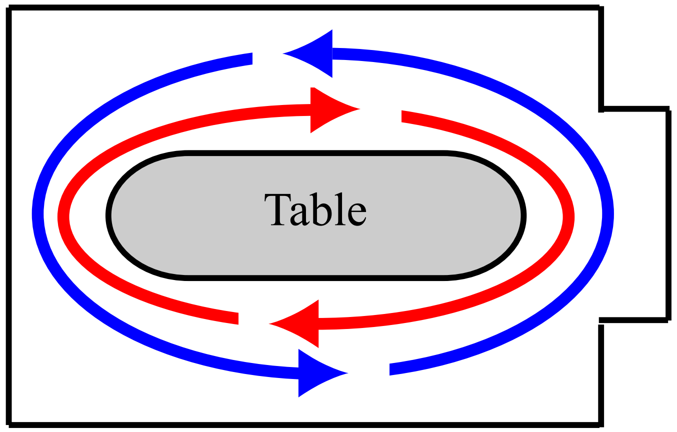
- online+random: 100k features, 11k frames, limit = avail. data
- offline: 2k frames, compressed & subsampled, compute-limited



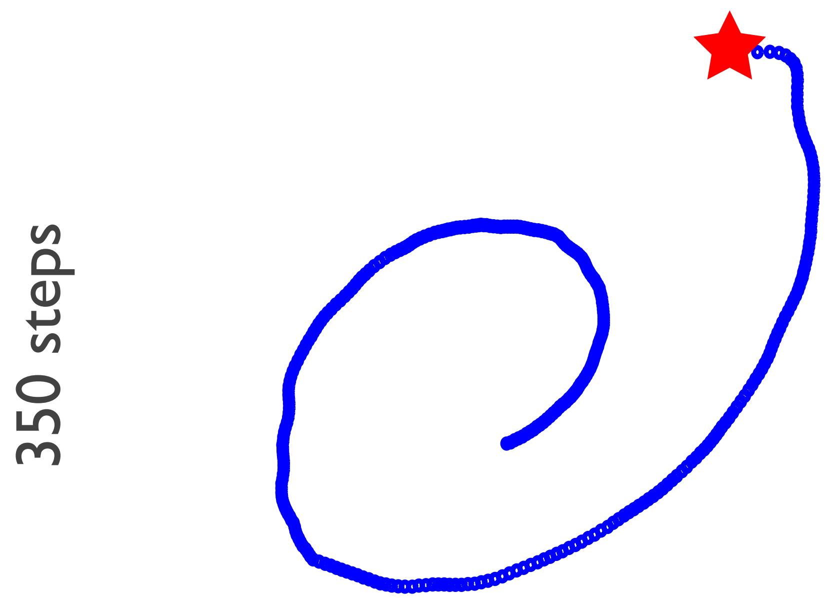
100 steps

Online Learning Example

Conference Room

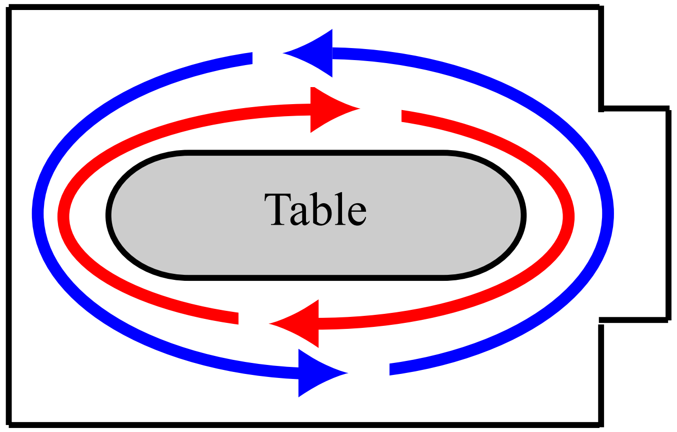


- online+random: 100k features, 11k frames, limit = avail. data
- offline: 2k frames, compressed & subsampled, compute-limited

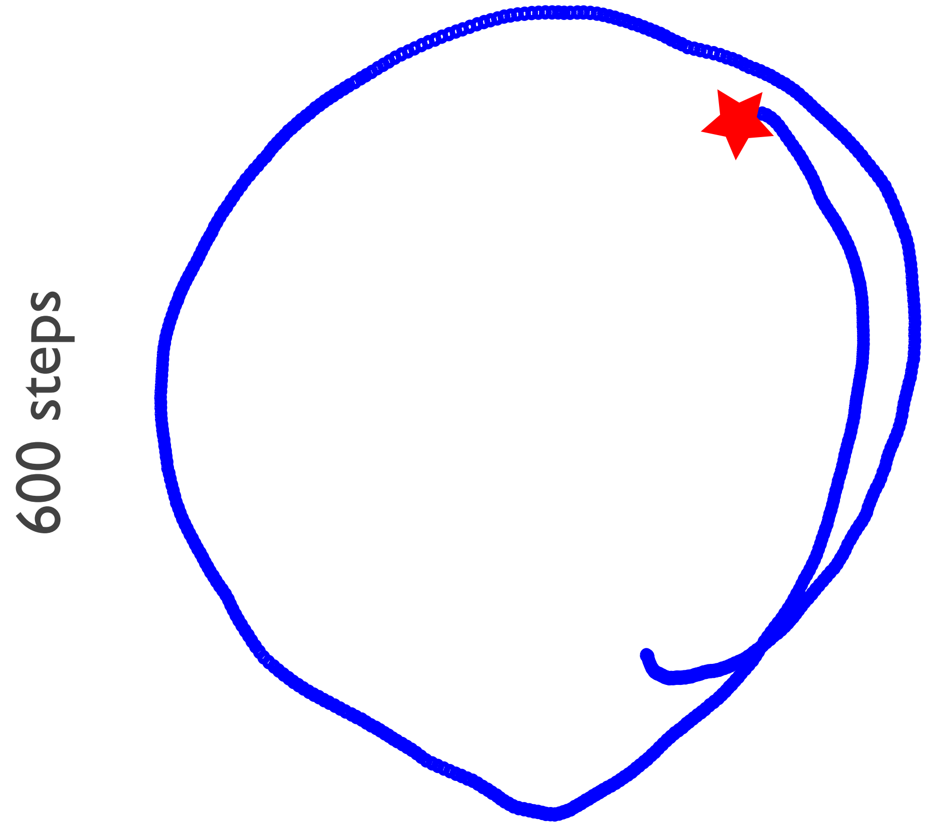


Online Learning Example

Conference Room



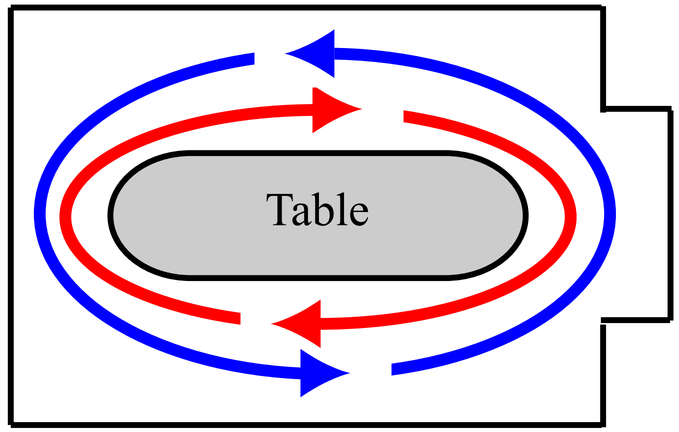
- online+random: 100k features, 11k frames, limit = avail. data
- offline: 2k frames, compressed & subsampled, compute-limited



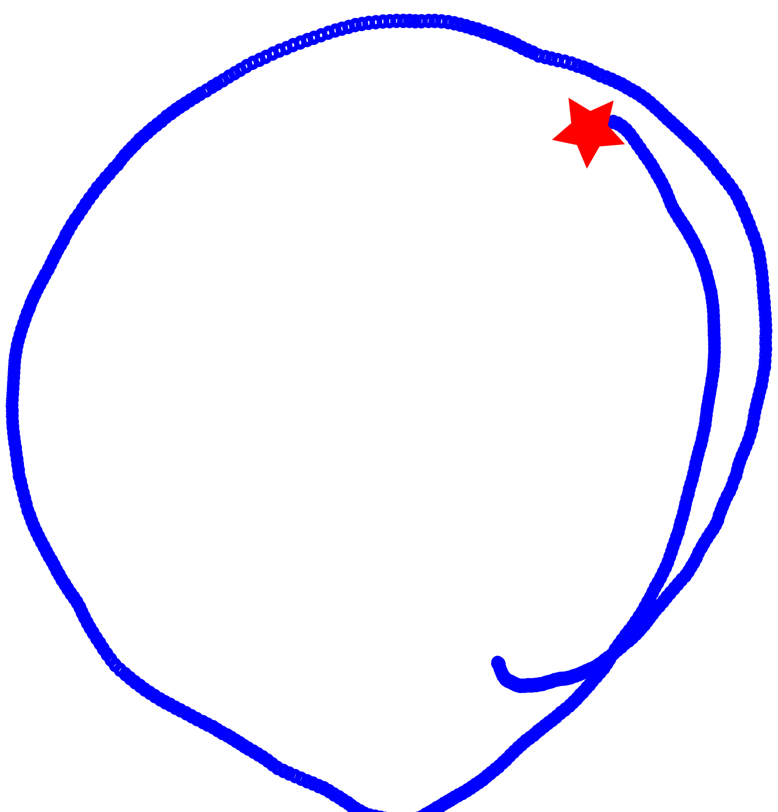
sense
learn
act

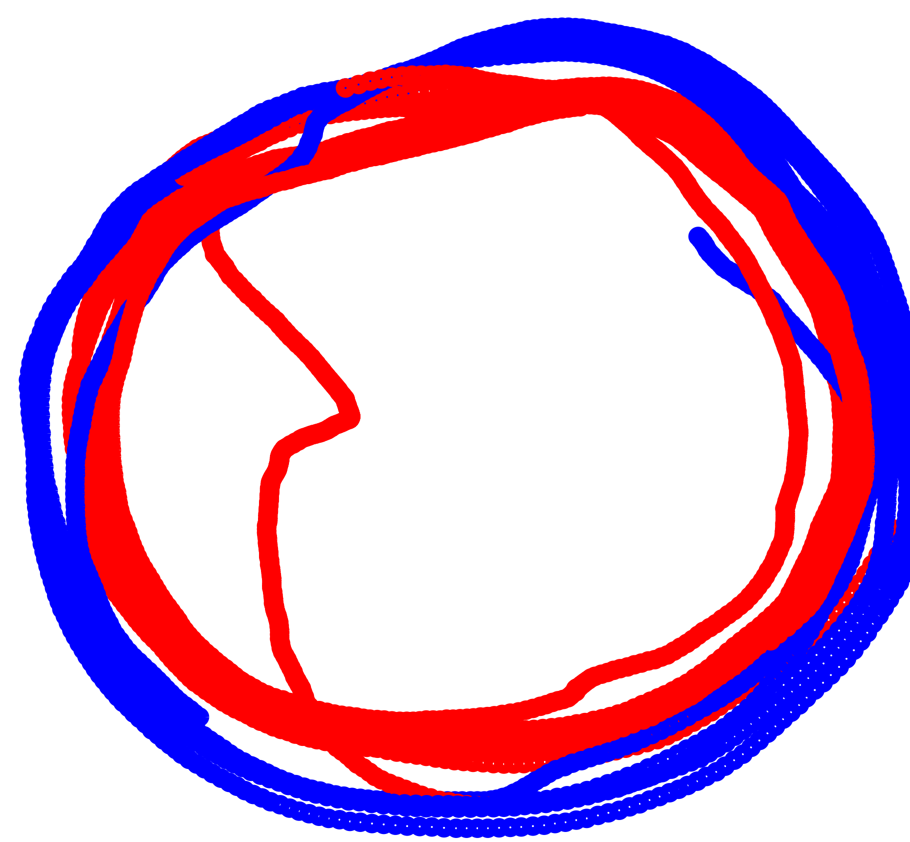
Online Learning Example

Conference Room



- online+random: 100k features, 11k frames, limit = avail. data
- offline: 2k frames, compressed & subsampled, compute-limited





sense
learn
act

Paper Summary

- We present spectral learning algorithms for PSR models of partially observable nonlinear dynamical systems.
- We show how to update parameters of the estimated PSR model given new data
    - efficient online spectral learning algorithm
- We show how to use random projections to approximate kernel-based learning algorithms